ICLR 2023 notable top 25%

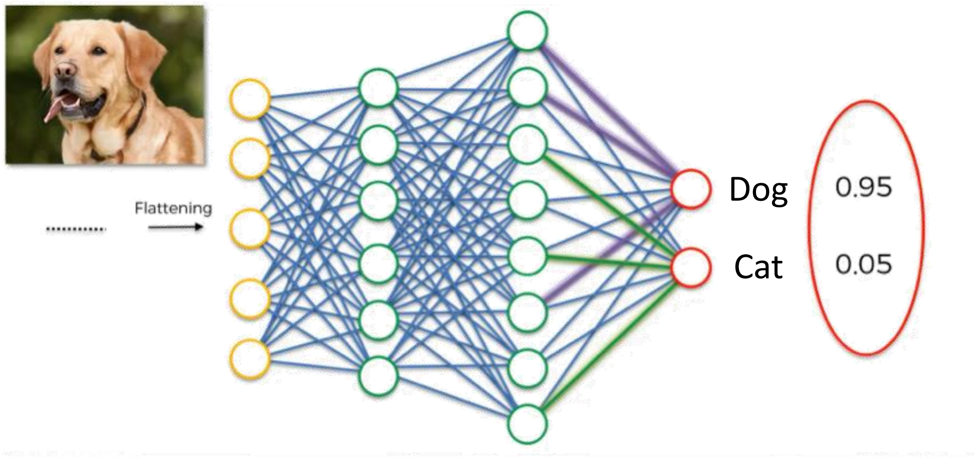
LEARNING LABEL ENCODINGS FOR DEEP REGRESSION

Deval Shah & Tor M. Aamodt

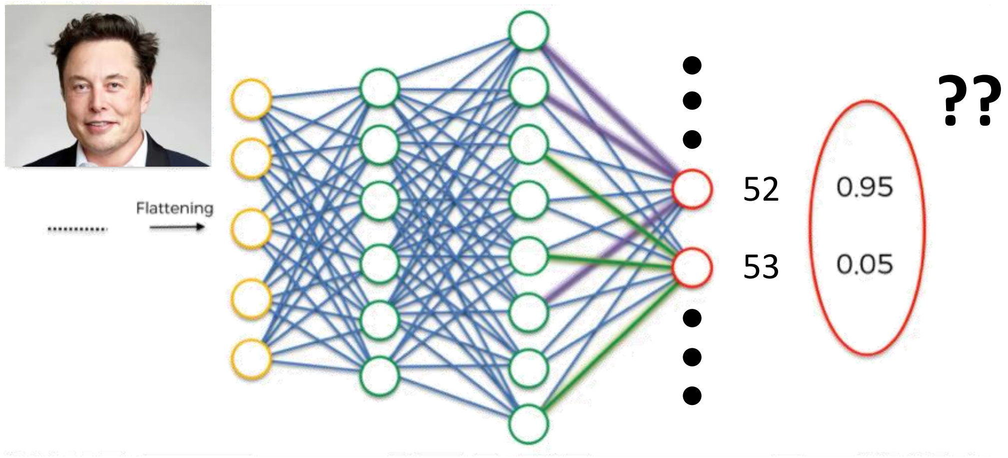
Department of Electrical and Computer Engineering
University of British Columbia, Vancouver, BC, Canada

Junwon Seo
EffL@POSTECH

Deep Regression

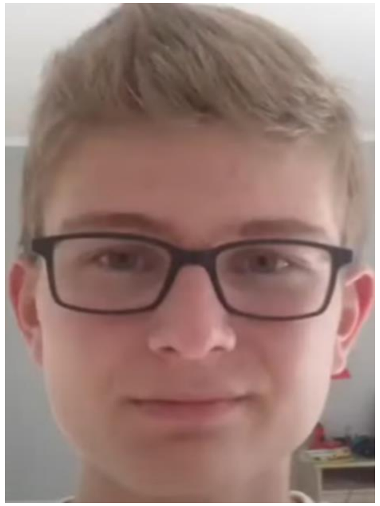


Classification



- Many real-world tasks involve continuous and even infinite target values
- In regression task, treating target value as distinct class is unlikely to yield best results.

Deep Regression




- In regression, it takes advantage of the similarity between people with nearby ages.
- Similar issues happen in medical application including heart rate, blood pressure, and oxygen saturation

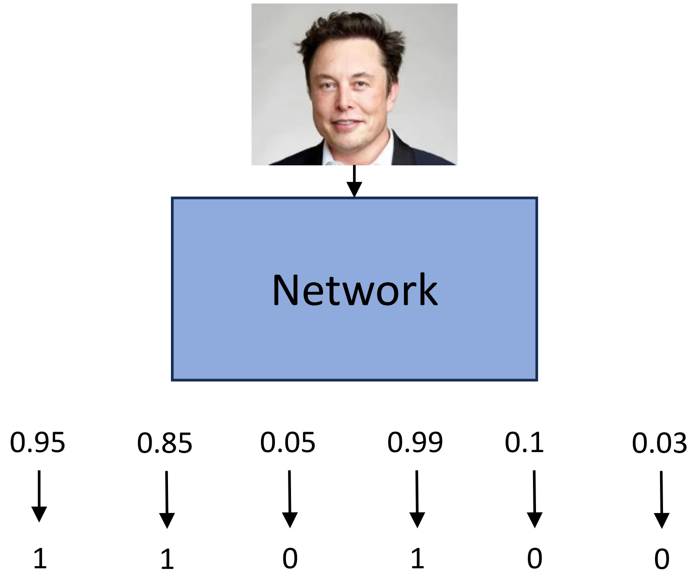
Deep Regression with Binary Classification

- Deep Regression has a problem that..
  - A typical generic approach performs poorly compared to task-specialized approaches.
  - Directly minimizing the MSE or MAE between targets and predictions

- Recently, generic approaches based on **regression by binary classification** have shown significant improvement.

- Regression by binary classification
  1. A real-valued label is quantized
  2. Converted to an M-bit binary code
  3. These binary-encoded labels are used to train M binary classifiers

Deep Regression with Binary Classification



- Example
  - The age of 52 would be 110100 using binary conversion as the encoder.
    (Training phase)
  - 110100 would be converted to real-value prediction using a decoding function (decimal convert)
    (Inference phase)
- This approach introduce ensemble diversity and error correcting capability.

Deep Regression with Binary Classification

Error Correcting Capability(ECC)

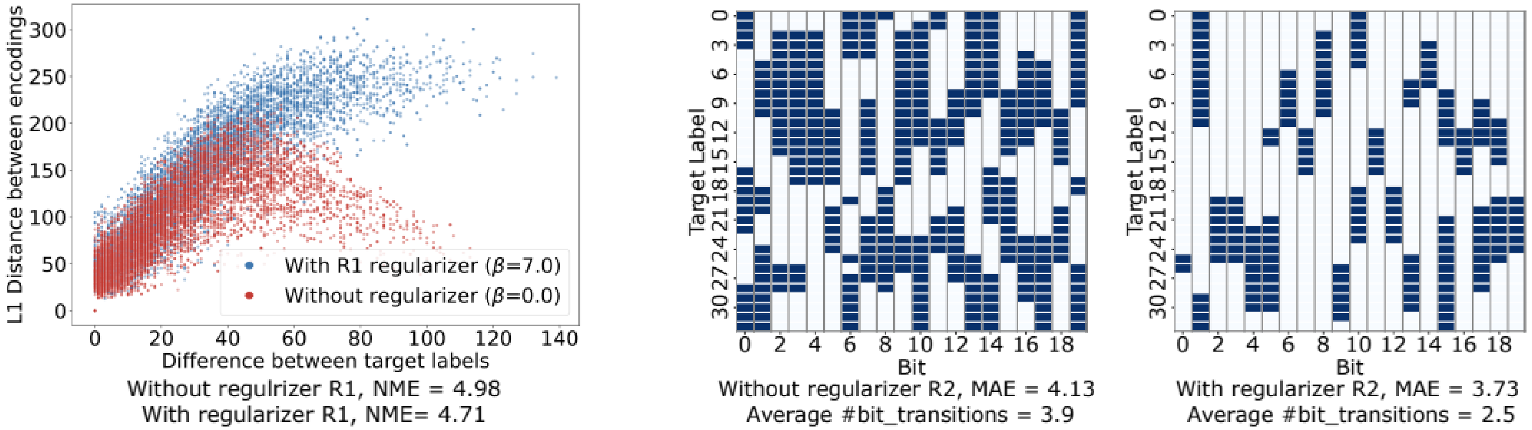
Target Label	Binary-encoded Label						
	B^7	B^6	B^5	B^4	B^3	B^2	B^1
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1
3	0	0	0	0	0	1	1
4	0	0	0	0	1	1	1
5	0	0	0	1	1	1	1
6	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1
8	1	1	1	1	1	1	1

(a) Unary encoding

- Exa
  -
  -
- This approach introduce ensemble diversity and error correcting capability.

How we find suitable label encoding for a given problem?

Factors of a good label encoding

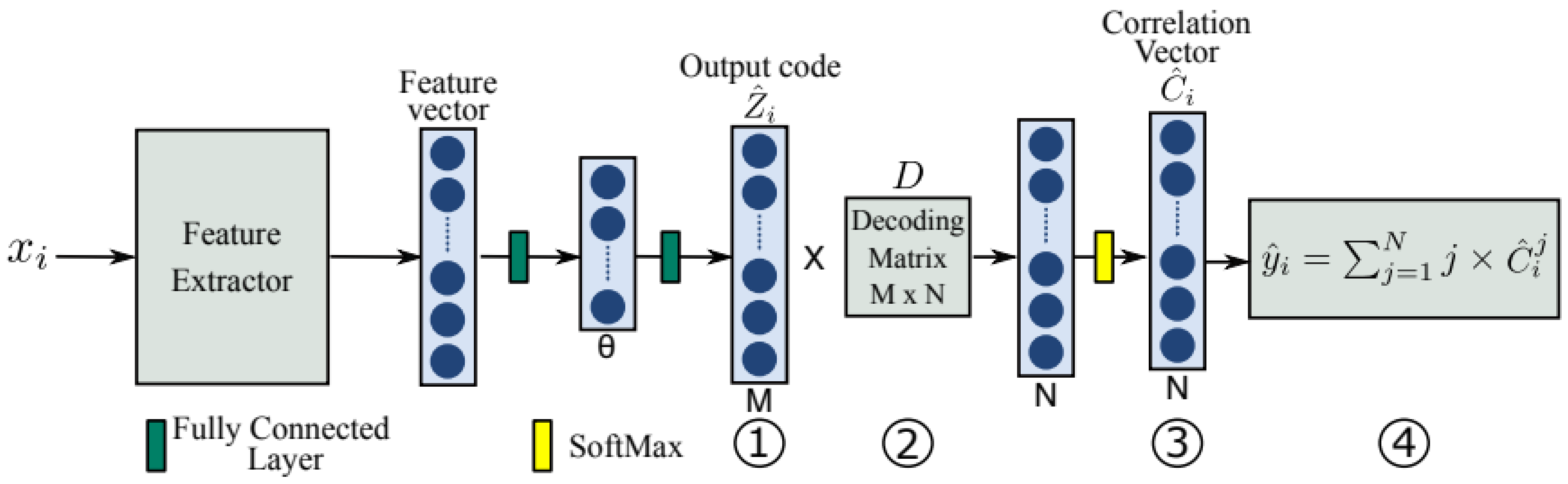


(a) Effect of regularizer R1 (b) Effect of regularizer R2

- Shah et al. (2022)* analyzed and proposed properties of suitable encodings for regression.
    1. Distance between learned encoded labels to be proportional to the difference between corresponding label values
    2. Reduce the complexity of a binary classifier's decision boundary by reducing the number of bit transitions

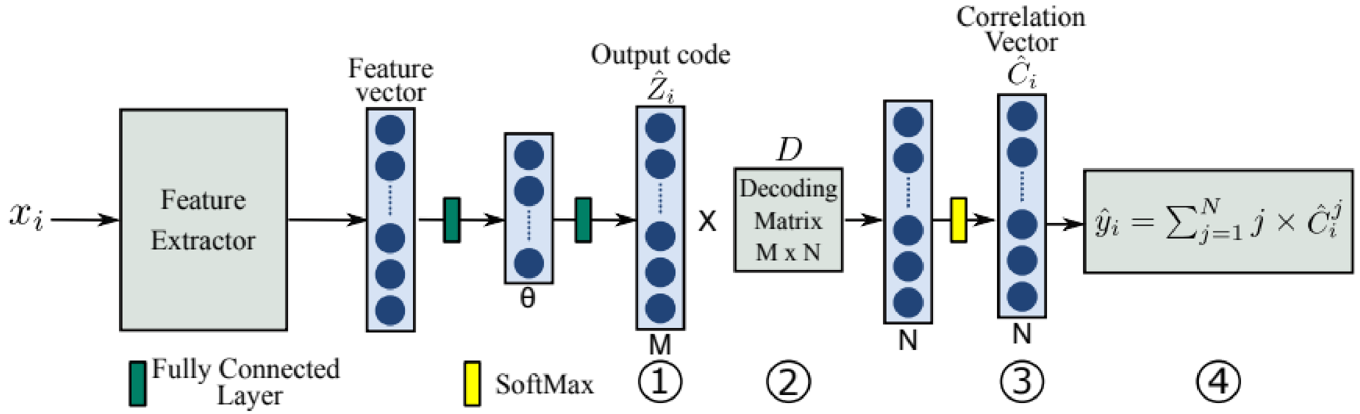
* Deval Shah, Zi Yu Xue, and Tor M. Aamodt. Label encoding for regression networks. In International Conference on Learning Representations

Regularized Label Encoding Learning (RLEL)



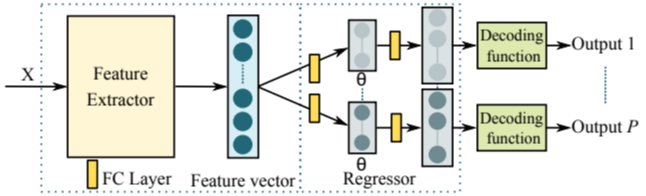
- An end-to-end approach to train **the network and label encoding** together
  - Relax the assumption of using discrete search space for label encodings.
    - Regularized search through a continuous space of real-valued label encodings
  - Enabling the use of continuous optimization approach.

Regularized Label Encoding Learning (RLEL)

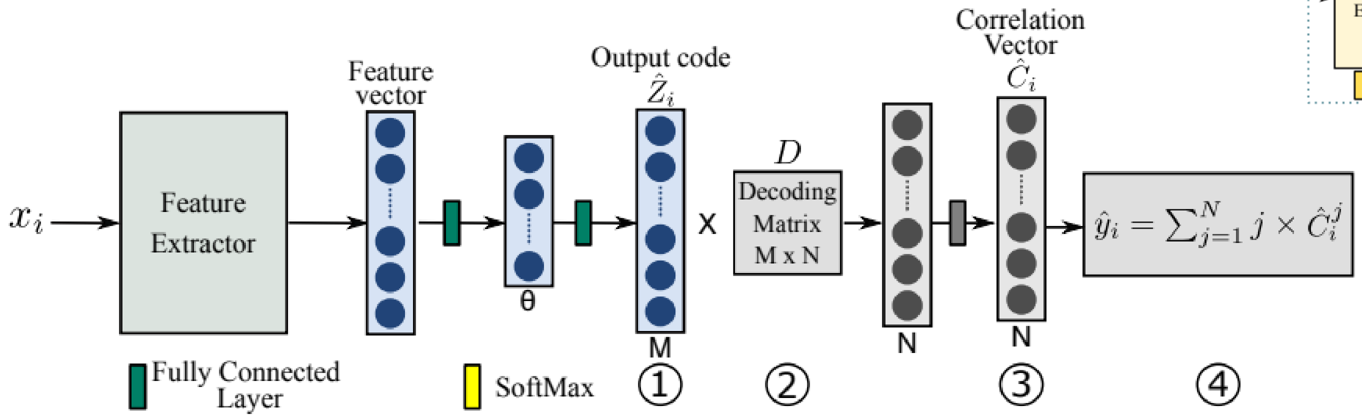


- x_i and y_i represent the input ant the real-valued target label for sample i.
  - For simplicity, $y_i \in [1, N]$ (scaled and shifted)
- $Q_i \in \{1, 2, \ldots, N\}$ represents the quantized target label.

Regularized Label Encoding Learning (RLEL)



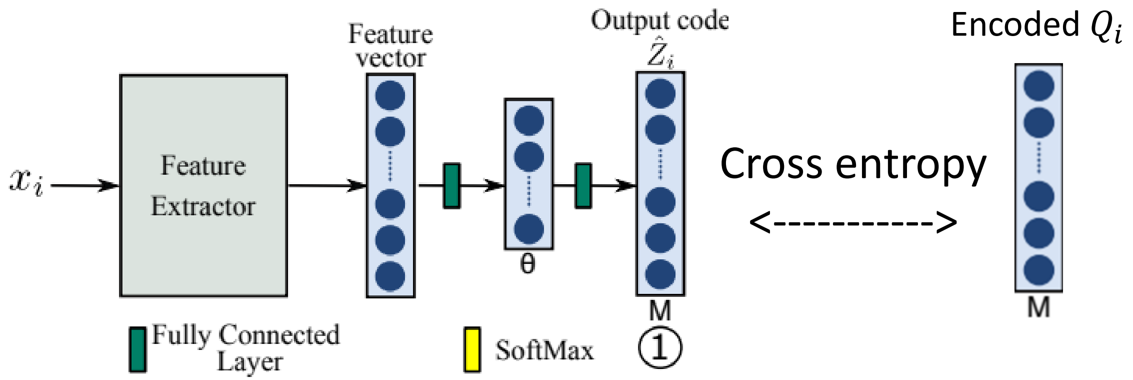
(b) Binary-encoded labels



- x_i is passed through a feature extractor and fully connected (FC) layers to generate the predicted encoding $\hat{Z}_i \in \mathbb{R}^M$
- An FC layer of size θ (θ < M) is added between the feature vector and output code.
  - This layer reduces the number of parameters in FC layers and improves accuracy (shown by previous work) (Shah et al., 2022)

* Deval Shah, Zi Yu Xue, and Tor M. Aamodt. Label encoding for regression networks. In International Conference on Learning Representations

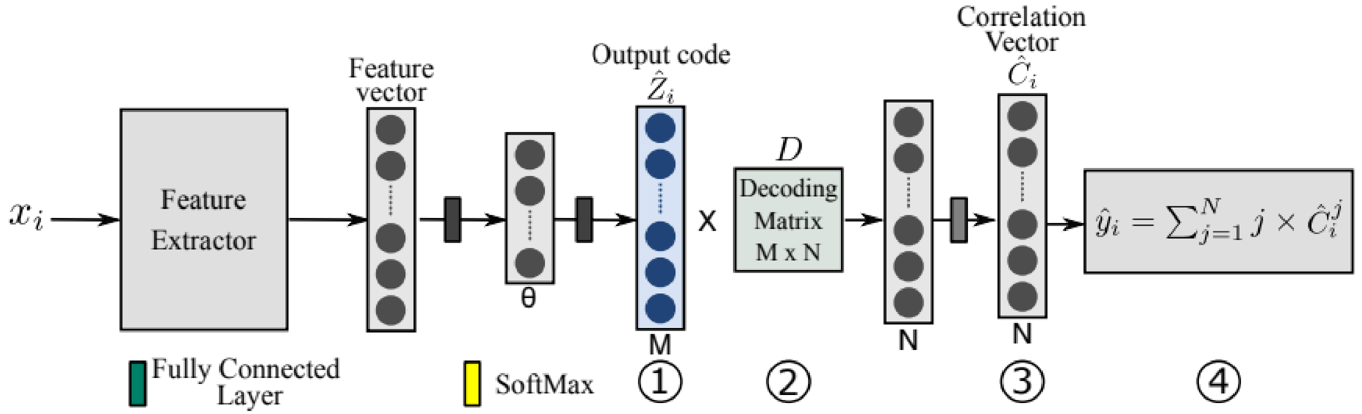
Regularized Label Encoding Learning (RLEL)



- Conventional methods are trained by comparing the encoded Q_i to the output code.

* Deval Shah, Zi Yu Xue, and Tor M. Aamodt. Label encoding for regression networks. In International Conference on Learning Representations

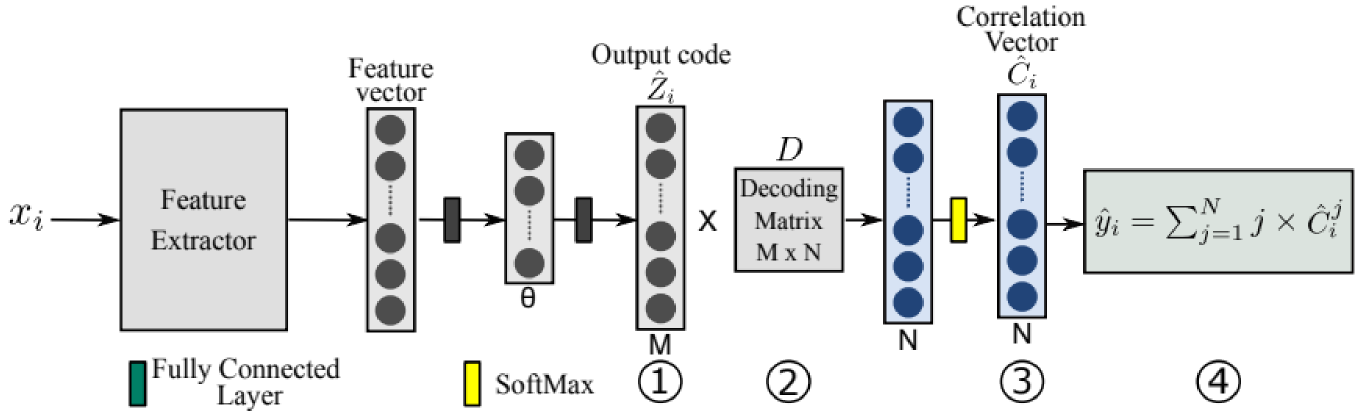
Regularized Label Encoding Learning (RLEL)



- Each neuron of the output code is a binary classifier, and the magnitude $\hat{Z}_i$ gives a measure of the confidence of the classifier-k
- The output code and a decoding matrix D $\in \mathbb{R}^{M \times N}$ are multiplied

* Deval Shah, Zi Yu Xue, and Tor M. Aamodt. Label encoding for regression networks. In International Conference on Learning Representations

Regularized Label Encoding Learning (RLEL)



- The output is passed through a softmax function to give a correlation vector $\hat{C}_i \in \mathbb{R}^N$
- $\hat{C}_i^k$ represents the probability that the predicted label y_i = k.

* Deval Shah, Zi Yu Xue, and Tor M. Aamodt. Label encoding for regression networks. In International Conference on Learning Representations

Regularized Label Encoding Learning (RLEL)

Target Label	Binary-encoded Label							
	B^8	B^7	B^6	B^5	B^4	B^3	B^2	B^1
1	1	1	1	1	1	1	1	1
2	1	0	1	0	1	0	1	0
3	1	1	0	0	1	1	0	0
4	1	0	0	1	1	0	0	1
5	1	1	1	1	0	0	0	0
6	1	0	1	0	0	1	0	1
7	1	1	0	0	0	0	1	1
8	1	0	0	1	0	1	1	0

$$E_{n,:} = \frac{1}{|\mathbb{S}_n|} \sum_{i \subset \mathbb{S}_n} \hat{Z}_i$$

- $\mathbb{S}_n$ represent the set of training samples with quantized target Q_i = n
- Encoder $\mathrm{E} \in \mathbb{R}^{N \times M}$
- $E_{n,}$ is s the encoding for target Q_i = n

- However, training the network solely with the loss between $\hat{y}_i$ and y_i does not constrain the search space of label encodings

RLEL – Regularizations

- R1: Distance between encodings
  - L1 distance between encodings for two labels should increase with the difference between two labels

$$||E_{i,:} - E_{j,:}||_1 \propto |i - j|$$

- R2: Regularizing bit transitions
  - **The number of bit transitions** in a bit-position of label encoding gives a measure of the binary classifier's **decision boundary's complexity**

$$\sum_{i=1}^{M} \sum_{j=1}^{N-1} |E_{j,i} - E_{j+1,i}|$$

RLEL – Regularizations : minibatch

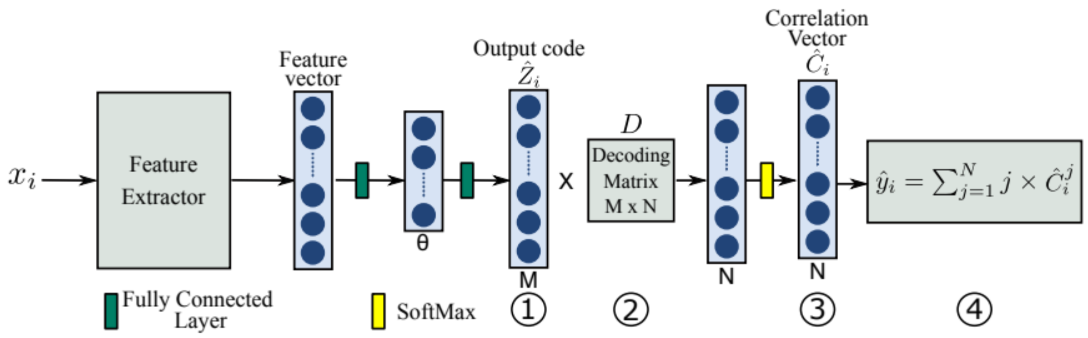
- Encoder E is measured from the output codes $\hat{Z}_i$ over the complete training dataset
  - However, deep neural networks are trained using mini-batches with K samples.
- R1: Distance between encodings

$$\mathcal{L}_1 = \sum_{i=1}^{K}\sum_{j=1}^{K} \max(0, 2 \times |y_i - y_j| - ||\hat{Z}_i - \hat{Z}_j||_1)$$

- R2: Regularizing bit transitions

$$\mathcal{L}_2 = \sum_{i=1}^{M}\sum_{j=1}^{N-1} |D_{i,j} - D_{i,j+1}|$$

RLEL – Loss function

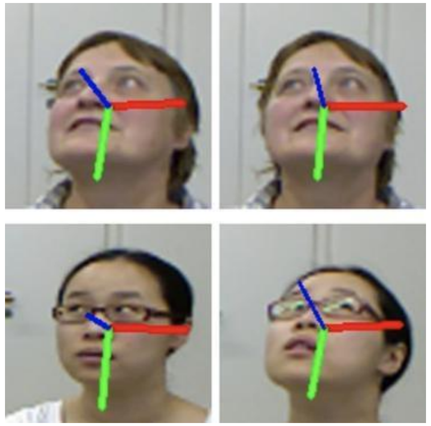


$$\mathcal{L} = \sum_{i=1}^{K} \mathrm{CE}(\hat{C}_i, \phi(y_i)) + \alpha \sum_{i=1}^{M} \sum_{j=1}^{N-1} |D_{i,j} - D_{i,j+1}| + \beta \sum_{i=1}^{K} \sum_{j=1}^{K} \max(0, 2 \times |y_i - y_j| - ||\hat{Z}_i - \hat{Z}_j||_1),$$

$$\text{where } \phi^j(y_i) = \frac{e^{-|j-y_i|}}{\sum_{n=1}^{N} e^{-|n-y_i|}} \qquad (5)$$

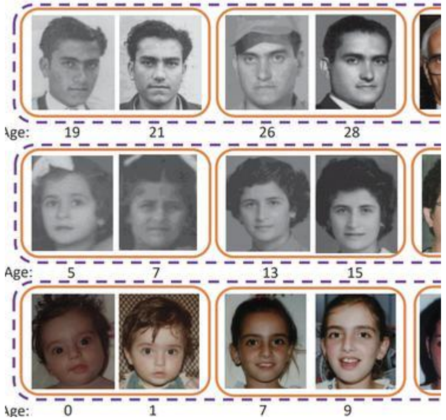
Experiments - Setup

Task	Feature Extractor	Dataset	Benchmark	Label range/ Quantization levels	θ
Landmark-free 2D head pose estimation	ResNet50 (He et al., 2016)	300LP (Zhu et al., 2016)/AFLW2000 (Zhu et al., 2016)	LFH1	0-200/200	10
		BIWI (Fanelli et al., 2013)	LFH2	0-150/150	10
Facial Landmark Detection	HRNetV2-W18 (Wang et al., 2020)	COFW (Burgos-Artizzu et al., 2013)	FLD1/FLD1_s (100%/10% training dataset)	0-256/256	10
		300W (Sagonas et al., 2013)	FLD2/FLD2_s (100%/10% training dataset)	0-256/256	10
		WFLW (Wu et al., 2018)	FLD3/FLD3_s (100%/10% training dataset)	0-256/256	10
Age estimation	ResNet50/ ResNet34	MORPH-II (Ricanek & Tesafaye, 2006)	AE1	0-64/64	10
		AFAD (Niu et al., 2016)	AE2	0-32/32	10
End-to-end autonomous driving	Pilot-Net(Bojarski et al., 2017)	PilotNet	PN	0-670/670	10



Head Pose Estimation

Facial Landmark Detection



Age Estimation

End-to-end Autonomous driving

Experiments - Results

	Error (MAE or NME)					
Approach	LFH1	LFH2	FLD1	FLD1_s	FLD2	FLD2_s
Simulated annealing	4.32±0.12	5.03±0.08	3.55±0.01	6.52±0.05	3.59±0.00	5.35±0.01
Autoencoder	**3.38**±0.01	4.84±0.02	3.39±0.01	4.85±0.03	3.39±0.00	4.20±0.05
LEL(w/o regularizers)	4.03±0.15	4.96±0.08	3.36±0.01	4.98±0.07	3.39±0.01	4.28±0.05
BEL(Manually designed)	3.56±0.11	**4.77**±0.05	**3.34**±0.01	**4.63**±0.03	3.40±0.02	**4.15**±0.01
RLEL	3.55±0.10	**4.77**±0.05	3.36±0.01	4.71±0.04	**3.37**±0.02	**4.15**±0.05

Approach	FLD3	FLD3_s	AE1	AE2	PN
Simulated annealing	4.52±0.02	6.38±0.01	2.33±0.01	3.17±0.01	4.25±0.01
Autoencoder	4.36±0.01	5.62±0.01	2.29±0.00	3.19±0.01	4.49±0.04
LEL(w/o regularizers)	**4.35**±0.02	5.68±0.04	2.30±0.01	3.17±0.01	3.22±0.02
BEL(Manually designed)	4.36±0.02	5.62±0.00	**2.27**±0.01	**3.11**±0.00	3.11±0.01
RLEL	**4.35**±0.01	**5.58**±0.01	2.28±0.01	3.14±0.01	**3.01**±0.03

- SA/AE do not optimize the encodings end-to-end with the regression problem.
- Error of LEL increases for smaller datasets, which suggests that RLEL generalize better
- Main objective of RLEL is to automatically learn label encoding
  - BEL need human intervention to design codes and multiple training runs.

Conclusion

- Analyze **properties of suitable encodings** in the continuous search space
- Propose **regularization functions for end-to-end learning** of **network parameters** and **label encoding**.
- Evaluate the proposed approach on **11 benchmarks and show significant improvement** over different encoding design methods and generic regression approaches.